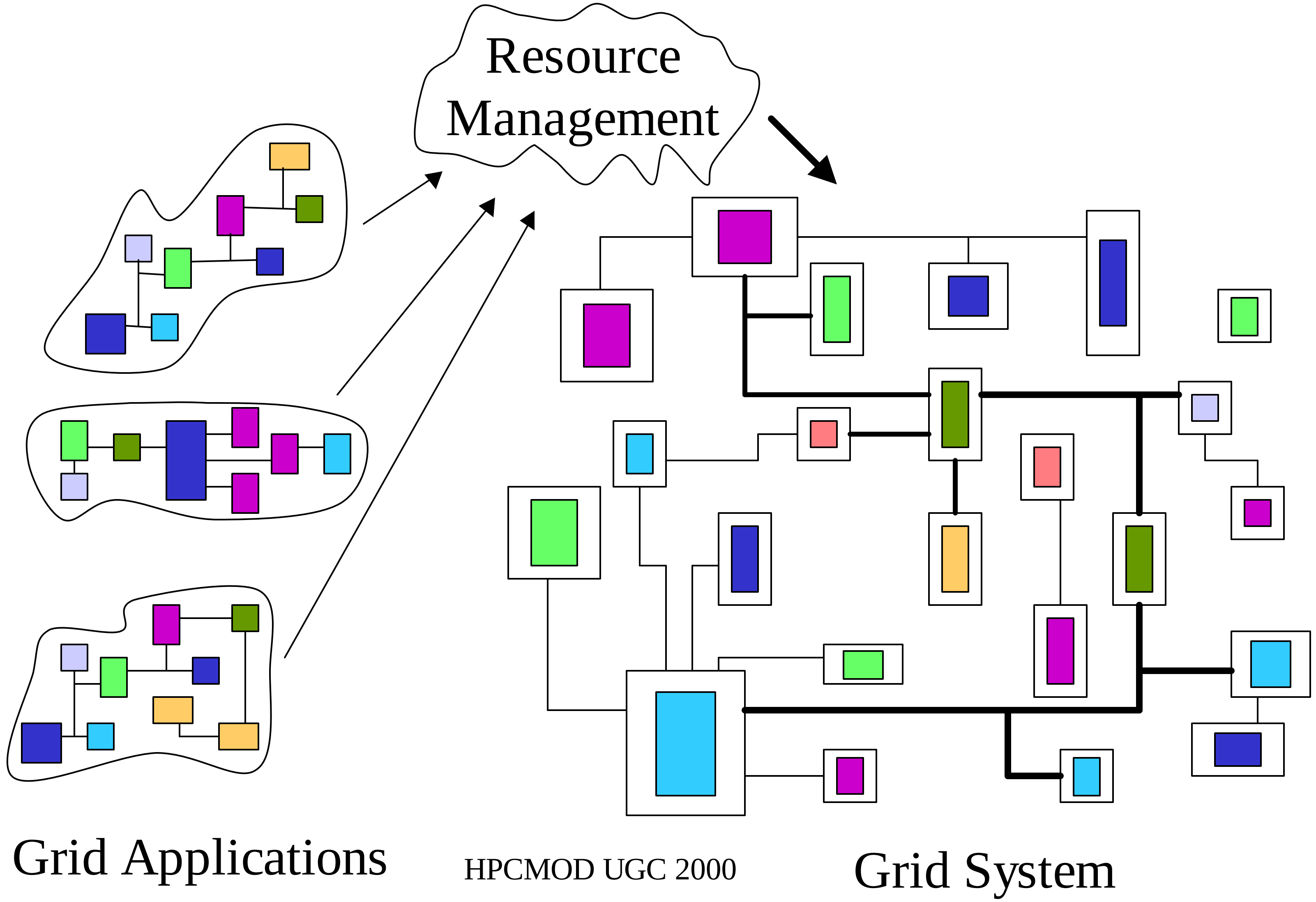
Research in
Multi-Resource Aware Scheduling Algorithms*

HPCMOD UGC, June 6th, 2000

William Leinberger, George Karypis, and Vipin Kumar
{leinberg, karypis, kumar} @ cs.umn.edu
Army High Performance Computing Research Center
University of Minnesota

* Research supported by NASA/AHPCRC/MSI

Scheduling Grid Applications: The Vision



Scheduling Grid Applications: Application-Centric Approach

- An application-specific scheduler is designed to:
  - Query the grid for resource candiates
  - Select the "best" resources, typically through advance reservation
  - Montior the resources during execution and adjust as necessary.

\+ Potentially better execution performance for the single application.

– Must be maintained along with the application

– Degrades to First-Come-First-Served resource allocation, resulting in low utilization of critical grid resources.

Scheduling Grid Applications: System-Centric Approach

- System level scheduling:
  - Accepts resource requests from grid-applications
  - Allocates grid resources to all waiting jobs to make best use of all grid resources.

\+ More efficient use of critical grid resources.

\- Much harder to implement general system level scheduler due to scale of grid systems.

Our Research focuses on system level scheduling approaches!

Scheduling Grid Applications: Assumptions

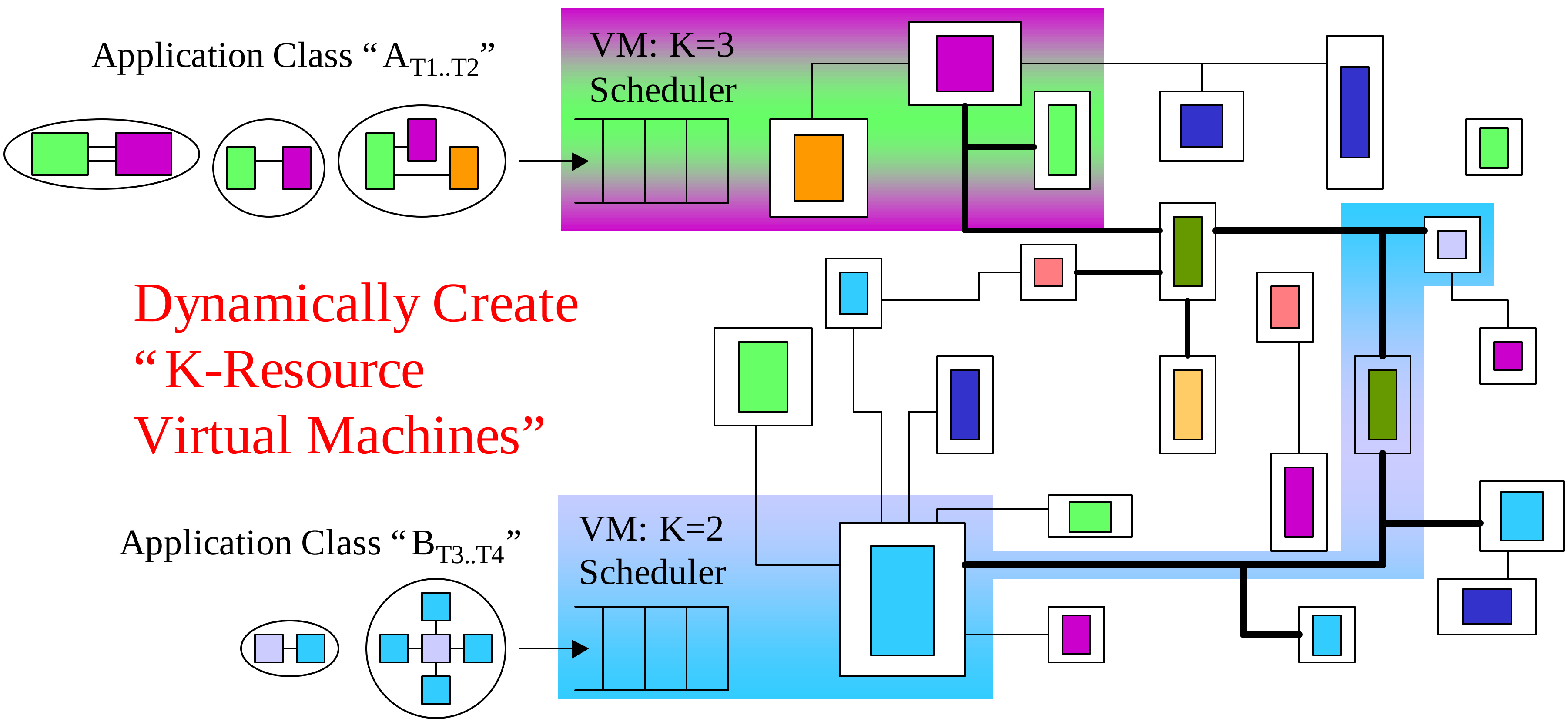
- Multiple instances of grid applications are submitted in common temporal space:
  - Parameter sweep studies
  - Real-time data collection and analysis
- Grid applications may be grouped into special resource configurations:
  - Functionally Related Resources: e.g. a data source, compute engine, and visualization interface.
  - Sparse Resources: e.g. peta-flop systems or data repositories, special purpose processor, high-speed data link
  - Atomic Resources: e.g. sensors, microscopes, etc.

We propose the use of dynamically created Virtual Machines!

Scheduling Grid Applications: A Solution

Grid Applications

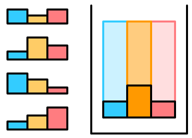
Grid System



Dynamically Create
"K-Resource
Virtual Machines"

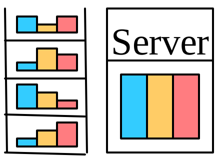
Research Topics for Scheduling K-Resource Virtual Machines

Bin Packing



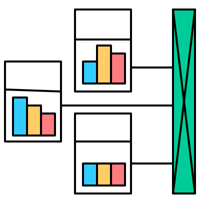
- The k-Capacity Bin Packing Problem
  - The Resource Balancing Heuristic

Scheduling



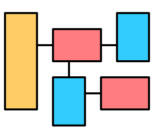
- Scheduling on a Single K-Resource VM
  - Jobs are items, free resources are a bin

Multi-Server



- Load Balancing Across Multiple Near-Homogeneous K-Resource VMs
  - In case one VM isn't enough

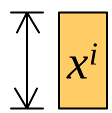
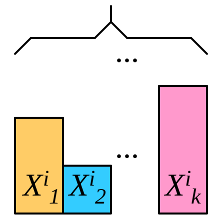
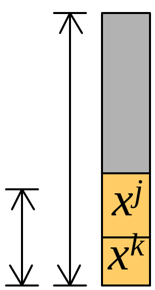
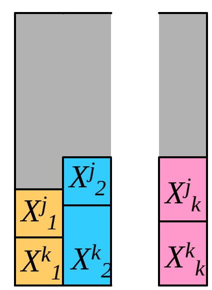
Multi-Resource



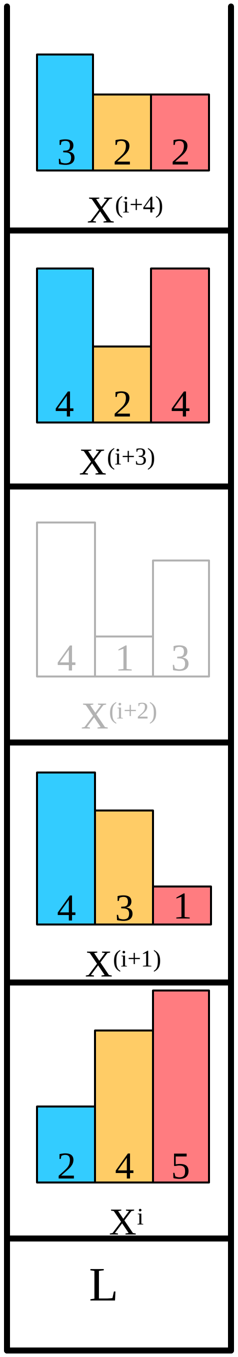
- Dynamic Creation of K-Resource VMs in a Computational Grid
  - How and Who

Bin Packing

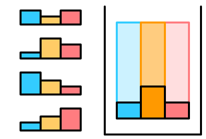
K-Capacity Bin Packing: Notations

	Single Capacity		k-Capacity	
Item List L:	$\{x^1, x^2, \ldots x^i, \ldots x^n\}$		$\{\mathbf{X}^1, \mathbf{X}^2, \ldots \mathbf{X}^i, \ldots \mathbf{X}^n\}$	
Item:	Size(x^i):		Sum, Max, Lex. $(X^i_1, X^i_2, \ldots, X^i_k)$:	
Bin:	Capacity C:		$\mathbf{C} = (C_1, C_2, \ldots, C_k)$:	
	Content B^j:		$\mathbf{B}^j = (B^j_1, B^j_2, \ldots, B^j_k)$:	

$\Rightarrow$ Goal: Minimize the number of bins required to hold all input items

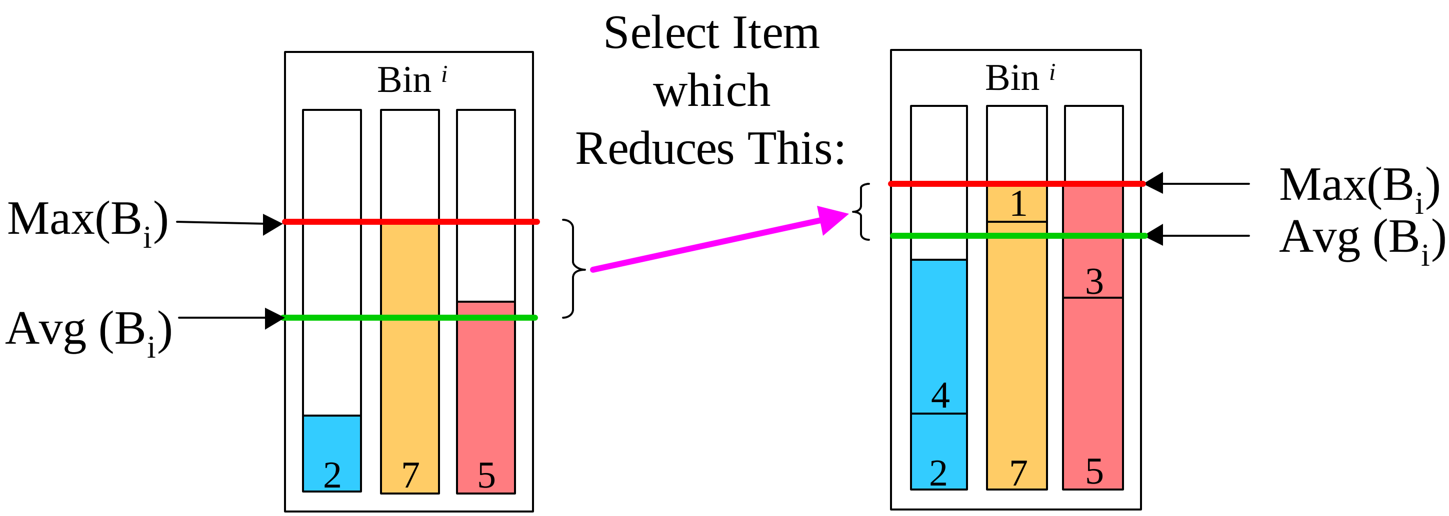


Bin Packing



K-Capacity Bin Packing: The Resource Balancing Heuristic

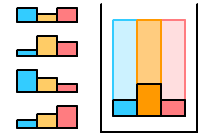
If Bin state is this: Bin State becomes:



$B_0 \leq B_2 \leq B_1$ $B_0 \approx B_1 = B_2$

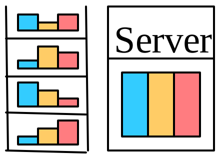
Note:

• $\mathrm{Max}(B_i)/\mathrm{Avg}(B_i)$ is an approximation to resource balancing

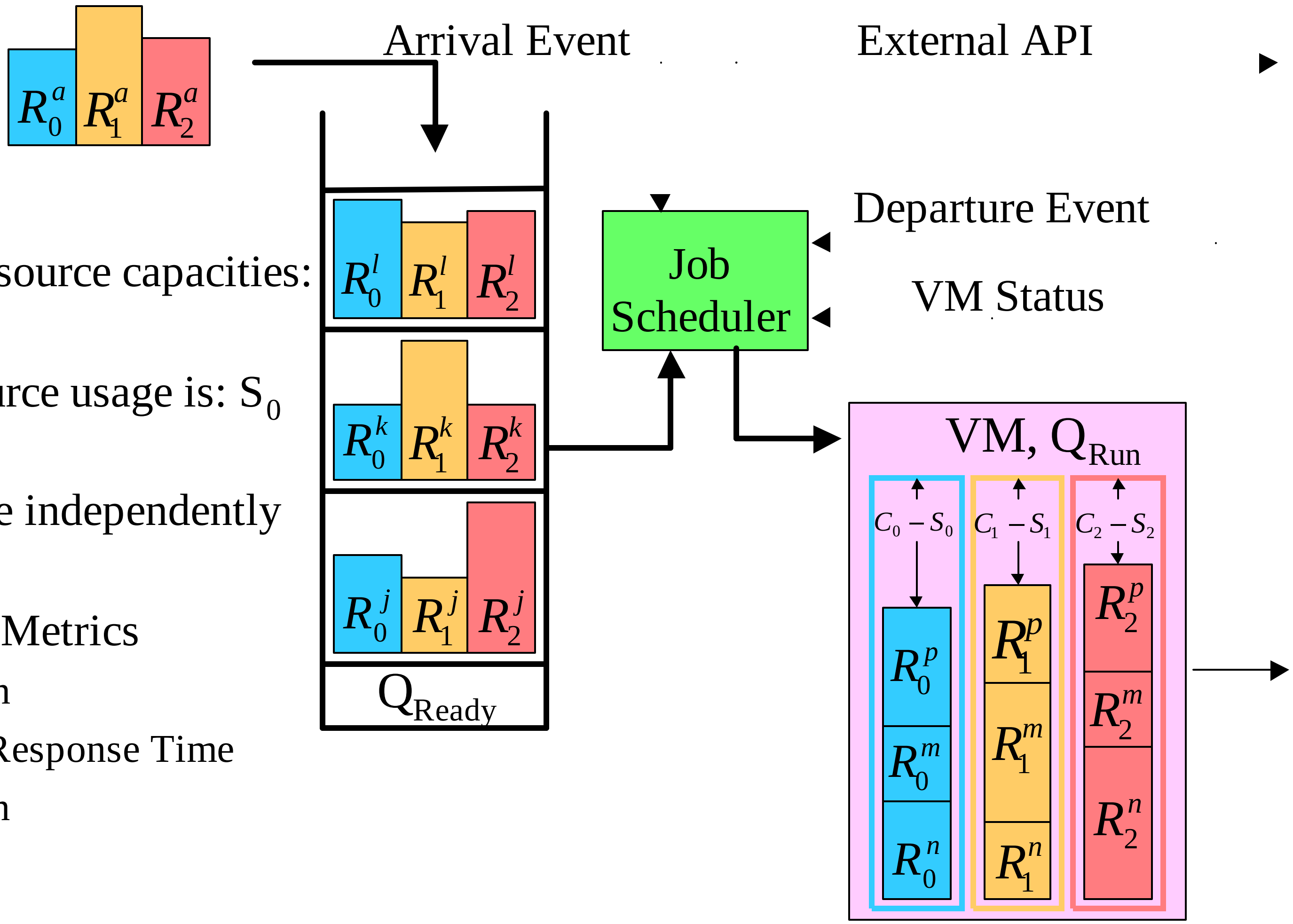


K-Capacity Bin-Packing: Status

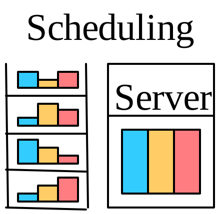
- Progress to Date:
  - Implemented various methods for approximating the Resource Balancing Heuristic.
  - 10-20% reduction in number of bins over First/Best-Fit variants.
  - Results published in ICPP’99.
- Future Work:
  - Item list pre-processing
    - Sorting by size
    - Combining complementary items
  - Dealing with large items and small bins



K-Resource System Scheduling: Notations



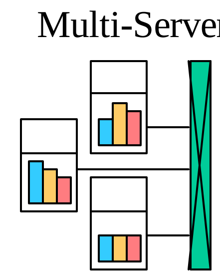
- VM has K resource capacities: $C_0 \ldots C_{K-1}$
- Current resource usage is: $S_0 \ldots S_{K-1}$
- Resources are independently allocatable
- Performance Metrics
  - Utilization
  - Average Response Time
  - Slowdown

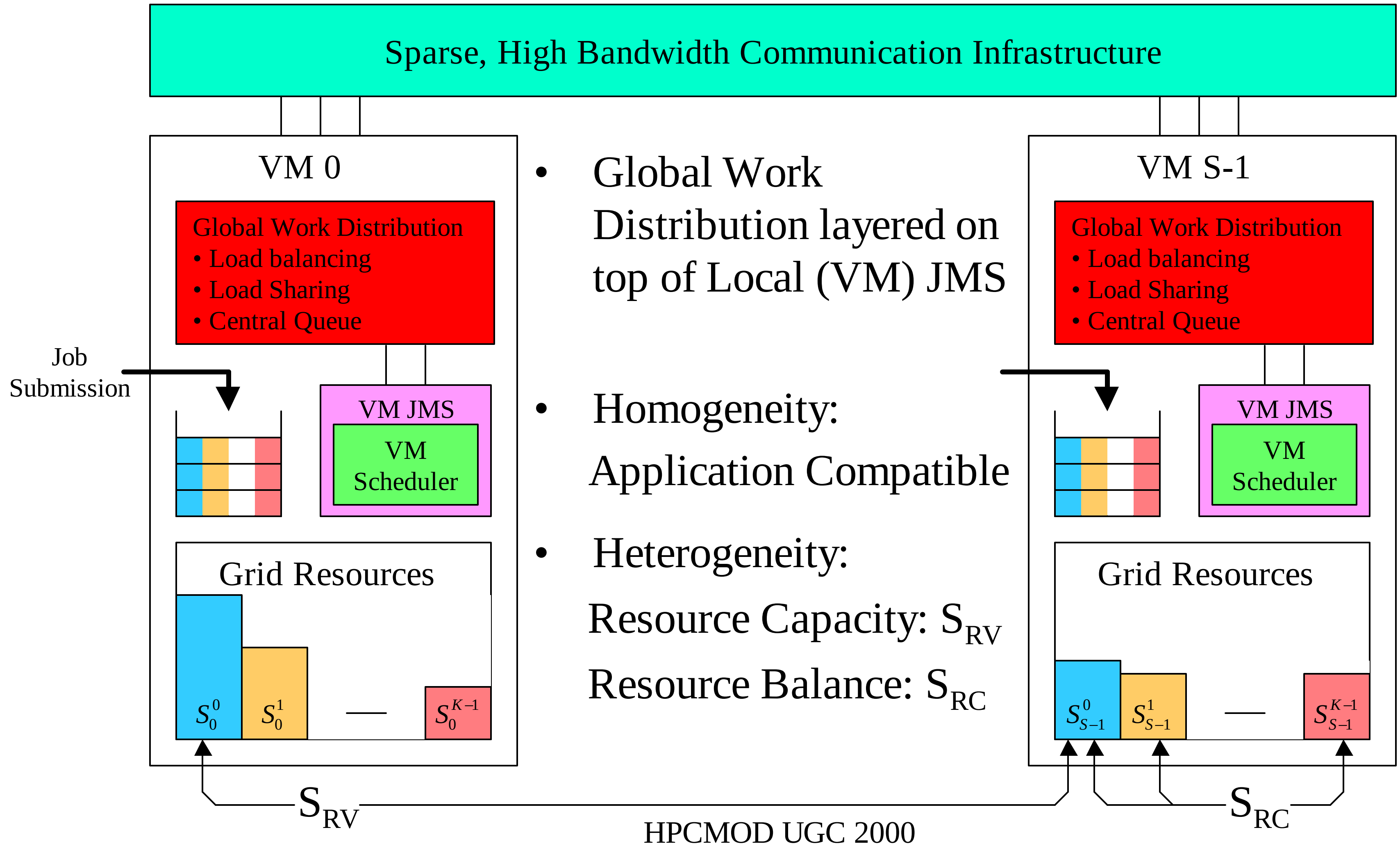


K-Resource Server Scheduling: Status

- Progress to Date:
  - Resource balancing heuristics are used to select jobs which best match remaining VM resources.
  - Designed new K-Resource aware backfill job selection algorithms.
  - 5-10% gains in Resource Utilization over EASY with First/Best-Fit, on synthetic k-resource workload.
  - 25-50% gains in Average Wait Time over EASY with First/Best -Fit.
  - Results published in SC’99.
- Work in Progress:
  - Unrestricted job queue re-ordering
  - Interaction between the K-Resource grid scheduler and the K single resource local schedulers.

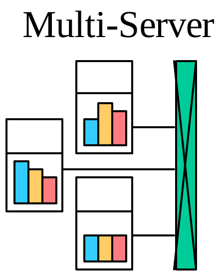
Load Balancing Across Near-Homogeneous K-Resource VMs: Notations





- Global Work Distribution layered on top of Local (VM) JMS
- Homogeneity:
  Application Compatible
- Heterogeneity:
  Resource Capacity: S_{RV}
  Resource Balance: S_{RC}

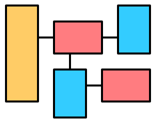
Load Balancing Across Near-Homogeneous K-Resource Servers: Notations



- Progress to Date:
  - Designed new load balancing policies, based on the resource balancing heuristic, which match the K-Resource capabilities of a server to the K-Resource requirements of the local workload.
  - Makes local scheduler (also K-Resource enabled) more efficient.
  - Achieved performance gains of 5-15% in throughput over classical load balancing techniques.
  - Results published in HCW 2000.
- Work in Progress:
  - Load balancing performance is consistently less than central queue.
  - Investigate compromise between central queue and pure load-balancing approaches.

Dynamic Creation of a K-Resource Virtual Machine in a Computational Grid

Multi-Resource



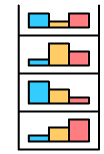
- How do you select resources?
  - Match the VM resource capabilities to the workload capabilities (Resource Balancing heuristic, again)
- Who's responsible?
  - Grid Infrastructure?
    - Cluster grid applications by resource requirement from a central grid "Ready Queue".
    - Create a K-Resource VM for each cluster/application class.
  - Problem Solving Environments?
    - PSEs used as interface between scientist and grid.
    - Typically multiple instances in use, generating application class specific workloads.
    - PSEs cooperate to make use of grid infrastructure co-allocation services when workload is sufficient to warrant a VM instantiation.

Summary

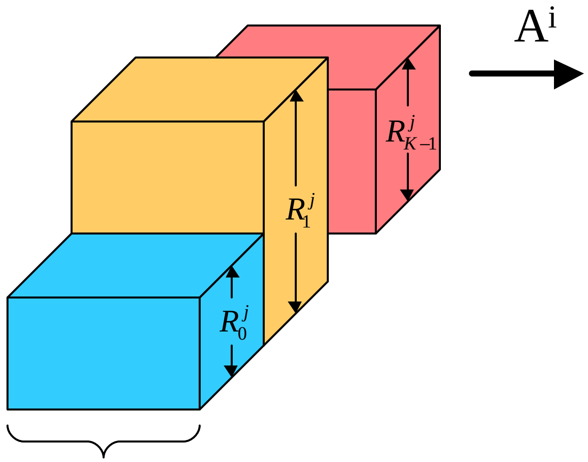
- We have proposed the use of K-Resource Virtual Machines to simplify the system-centric scheduling of multi-resource grid-scale applications.
- Our research is incrementally evolving a suite of K-Resource aware scheduling algorithms.
- The mechanisms for creating K-Resource Virtual Machines are extensions of current grid infrastructure services:
  - GLOBUS: A VM is a form of a co-allocation request.
  - LEGION: A VM is a composite object, created from the individual resource objects plus VM scheduler object.
- Still need more data on grid application workloads and system configurations to validate approach.

Backup Slides

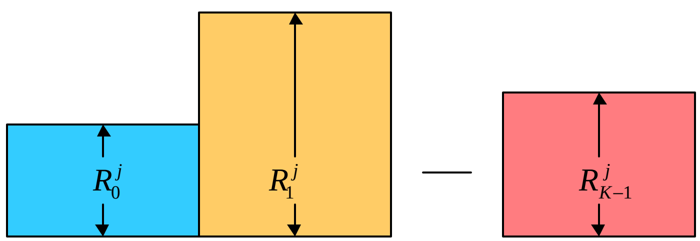
Workload



K-Resource Workload: Notations



Qualitative Notion of Job j



K-Resource Job Icon

- Job Assumptions:
  - Each job, j, has K critical resource requirements: $R^j_0 \ldots R^j_{K-1}$
  - All K resources are required simultaneously
  - The execution time is known, T^i
  - Jobs arrive at a rate A^i to a VM system S^i
  - Jobs are independent

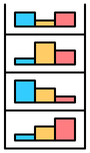
The K-Resource Workload Model: Status

- Progress to Date:
  - Created a synthetic model for a K-Resource workload
    - Simple extension of single resource (CPUs) parallel processing workload models
    - Inter-Job resource probability distribution
    - Intra-Job resource correlation
- Work in Progress:
  - Searching for characteristics of grid-scale application workloads
    - Resource configuration requirements for application classes, e.g. collaborative design environments, real-time experiment control, distributed parallel processing, etc..
    - Characteristics of resource classes, e.g. data source/sink, network link, compute engine, visualization, storage devices, etc..

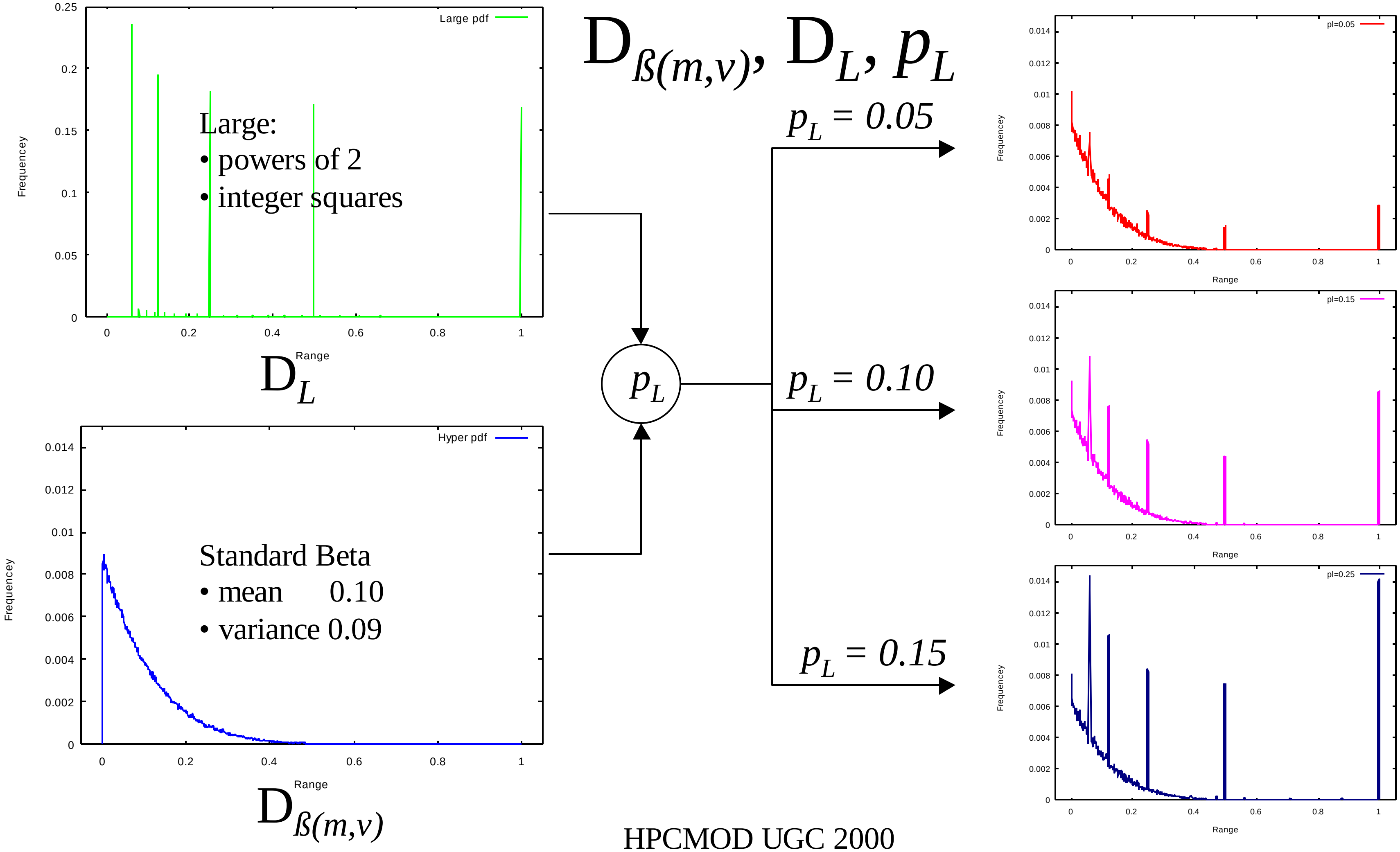
Relevant Research

- Job Classifications
  - Rigid, Evolvable, Modable, Malleable
- Workload Trace Analysis
  - General (size=CPUs)
    - Size distributions are exponential/hyperexponential
    - Large percentage from discrete sizes - powers of 2, squares of integers
    - Execution time is weakly correlated to job size
  - Memory
    - Many discrete sizes as well
    - Weakly correlated to number of CPUs
  - I/O
    - Input, Computation, Output Phases
    - Bursty

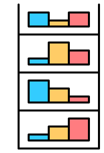
Workload



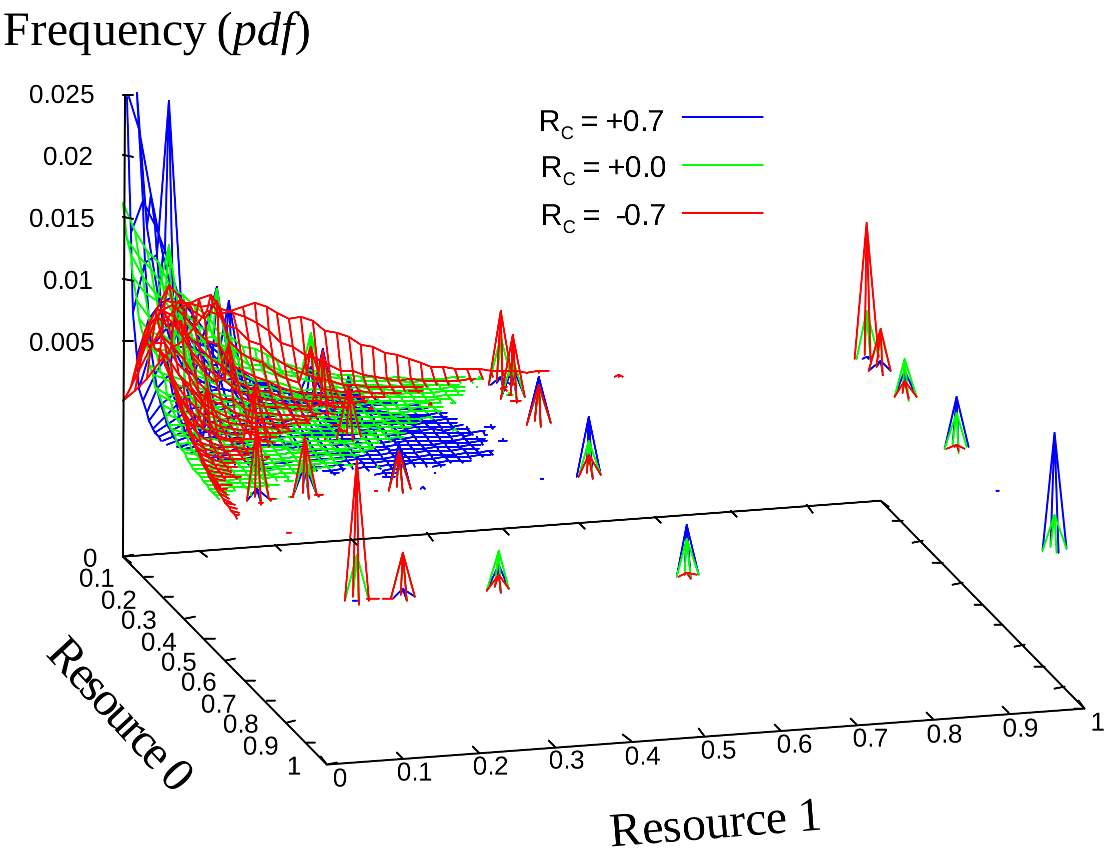
Initial Approach: Inter-Job Resource Distributions



Workload

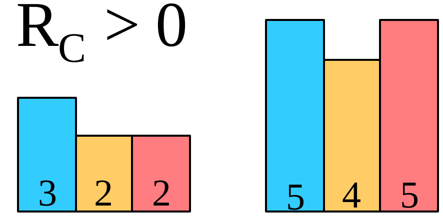


Initial Approach:
Intra-Job Resource Correlation, R_C

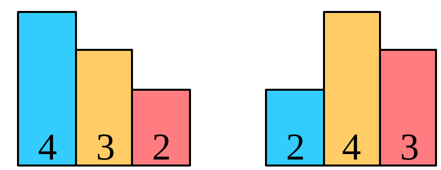


K=2, p_L=0.1, ß(0.10, 0.09)

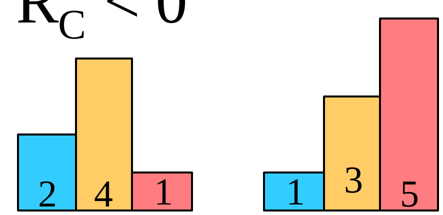
$R_C > 0$

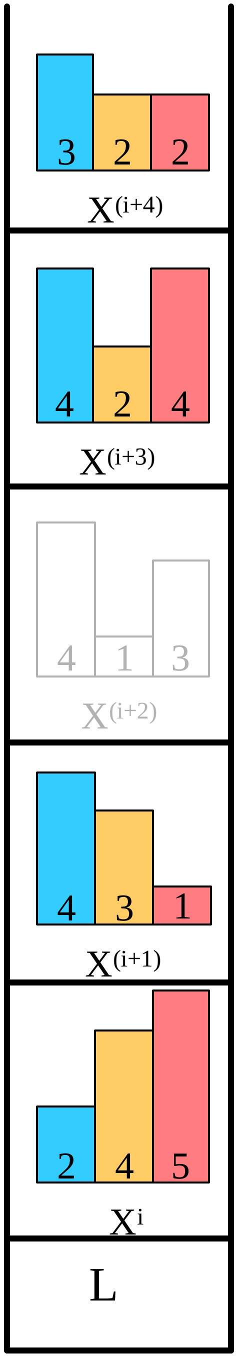


$R_C \sim 0$



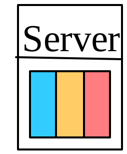
$R_C < 0$





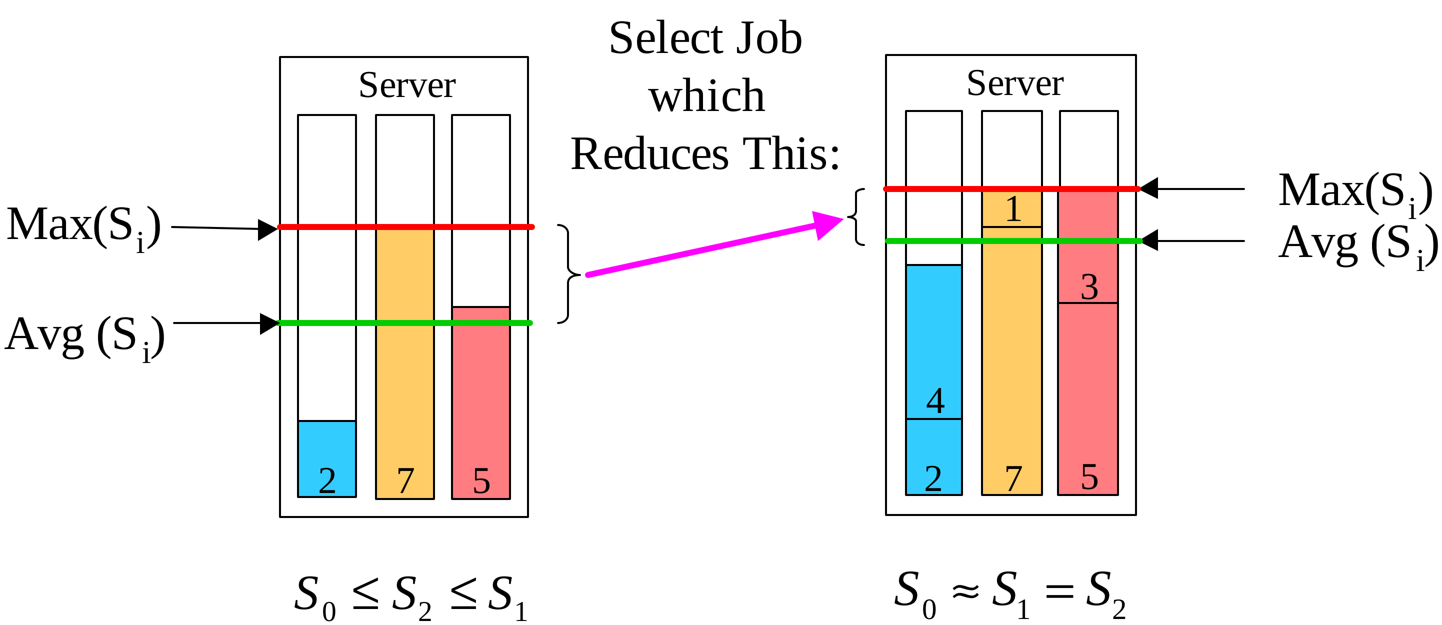
K-Resource VM Scheduling: Progress
The Resource Balancing Heuristic

Scheduling



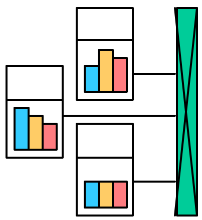
If VM state is this:

VM State becomes:



Note:

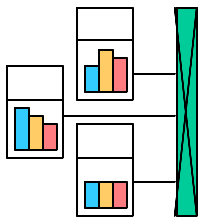
- Max(S_i)/Avg(S_i) is an approximation to resource balancing

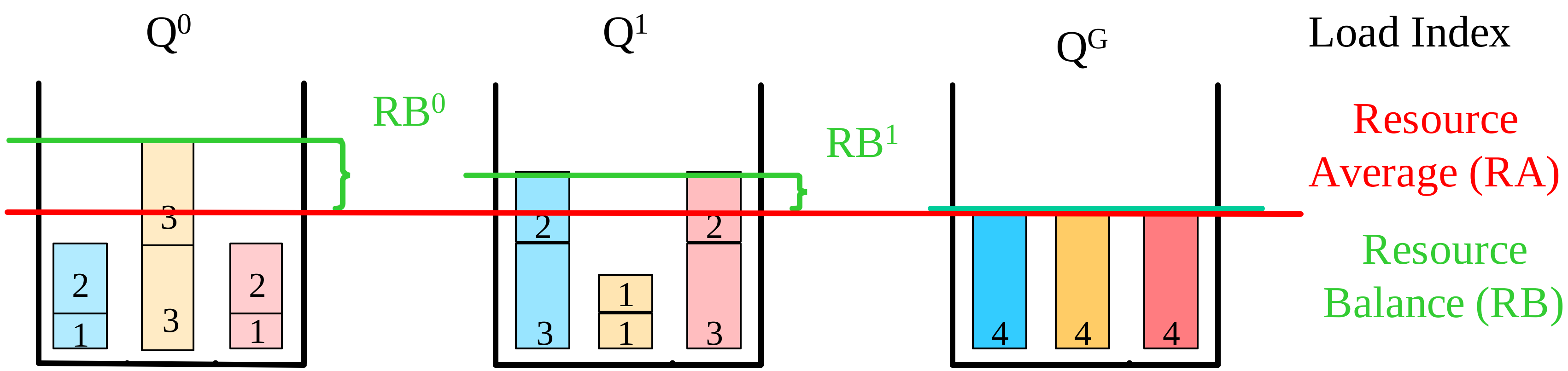


Components of Classical Load Balancing

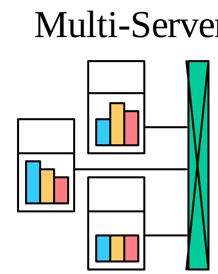
Policy	Description	Baseline Approach
Load Index	Describes Load on each Server	Job Queue Lengths, Resource Average (RA)
Information Policy	How and When to update Load Index	Global, Instantaneous
Transfer Policy	Triggering a load balancing action	Threshold-Initiated Sender, Receiver, and Symmetric
Selection Policy	Which job to move	Latest Job Arrived (LSP)
Location Policy	Where to move job from/too	Lowest Load, Highest Load

Limitations of Resource Average (RA) Load Index in the Presence of Multiple Resources

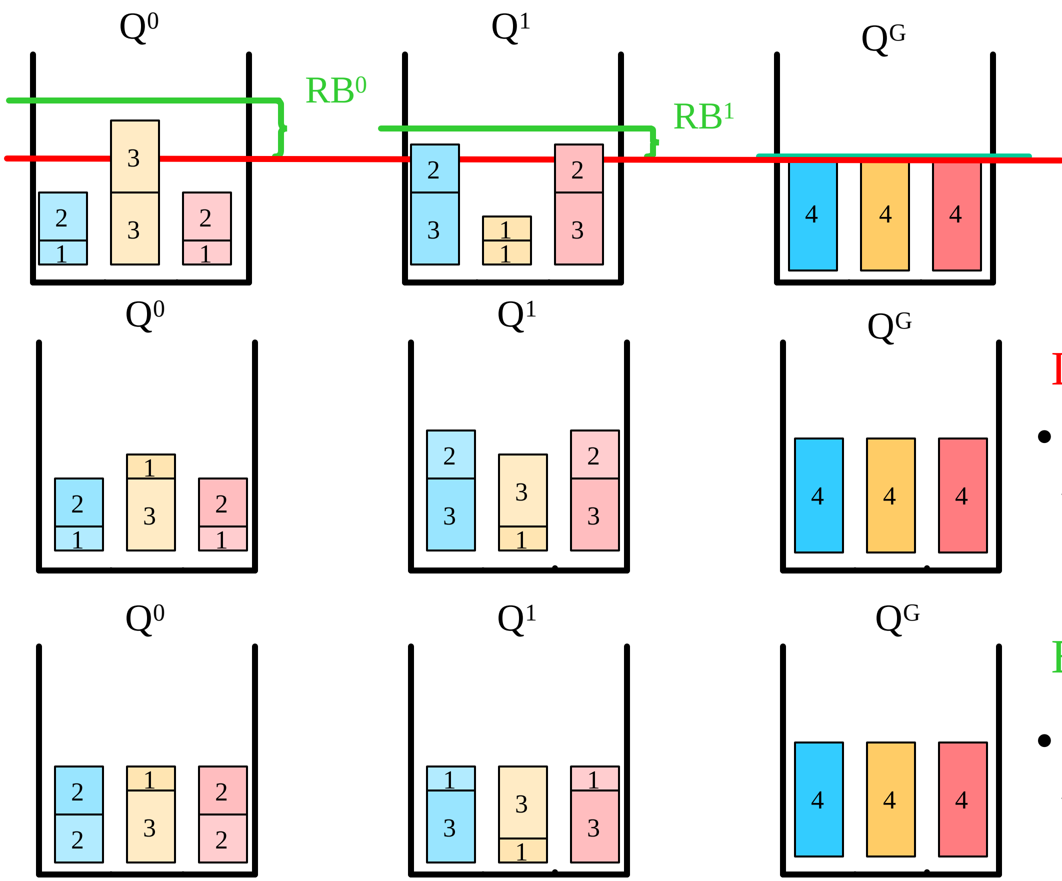




- The relative total resource requirements of the Workload should match the relative resource capabilities of the Server.
  - e.g. The workload of a server with a large memory configuration should contain mostly memory intensive jobs.
- Extend the classical Load Index, RA to RA+RB, to trigger load balance actions in the event of a workload resource balance mismatch.



Limitations of Latest Job (BSP) Selection Policy in the Presence of Multiple Resources



Resource Average (RA)

Resource Balance (RB)

Last Job Selection Policy (LSP)

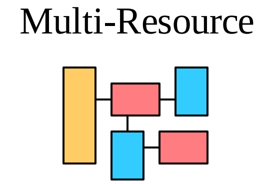
- Doesn't actively correct workload balance

Balanced Selection Policy (BSP)

- Actively corrects workload balance

- Add new job selection policy, BSP, to actively correct a workload resource balance mismatch.

Dynamic Creation of K-Resource Virtual Machines in a Computational Grid



- When Should a K-Resource VM be Created?
  - Upon submission of a single grid application? No!
    - Leads to First-Come-First-Served use of critical resources
  - Problem Solving Environments? Yes!
    - PSEs used as interface between scientist and grid
    - Typically multiple instances in use, generating application class specific workloads
    - PSEs cooperate to make calls to grid infrastructure VM services when workload is sufficient
- Creating a K-Resource VM in the Grid Infrastructure:
  - Grid Infrastructure
    - GLOBUS: Add service to create VM, similar to co-allocation request.
    - LEGION: Object to create a VM composite object.